CLAIMS

1. Method for verifying a transformer of a so-called source code into a so-called transformed code designed for an embedded system, said source and transformed codes being associated with virtual machines, characterized in that it comprises at least the following steps:

- determining, for each of said source (1) and transformed (3) codes, a first common subset (13), constituting a single virtual machine that factors in the behavior of these two codes (1,3);
- determining, for each of said source (1) and transformed (3) codes, a second subset (10, 30) constituted by a plurality of so-called auxiliary functions ($10_i - 30_i$) used by said single virtual machine, said auxiliary functions ($10_i - 30_i$) representing residual differences between said source (1) and transformed (3) codes;
- associating said auxiliary functions in pairs, a first auxiliary function (10_i) of each pair belonging to said second subset (10) associated with said source code (1) and a second auxiliary function ($30i$) of each pair belonging to said second subset (30) associated with said transformed code (3);
- verifying (6) a given correspondence property between said auxiliary functions ($10_i - 30_i$) of all of said pairs; and
- verifying that said transformation of the source code (1) into a transformed code (3) by said converter (2) satisfies said given correspondence property.

2. Method according to claim 1, characterized in that said correspondence property is a logical relation, so that said auxiliary functions of each of said pairs ($10_i - 30_i$), when executed, generate results linked by said logical relation, and in that this relation is the identity relation for so-called observable entities of each of said source and transformed codes, for any pair of auxiliary functions, so that the functionalities of said source code (1) are retained when said transformation into said transformed code (3), and said verification of the code transformer (2), are performed.

3. Application of the verification method according to either of claims 1 and 2 to a code transformer (2) that generates, from said source code (1), a transformed code (3) designed to be stored in memory means (71) of a chip card (7).

4. Application according to claim 3, characterized in that, said transformed code being a program written in the virtual machine of a given computer language, said chip card (7) is a chip card that stores a plurality of software applications (A_1 through A_n) written in this transformed code (3).

5. Application according to claim 3 or 4, characterized in that said source code (1) is a program written in the "JAVA" (registered trademark) virtual machine and said transformed code (3) is a program written in the "JAVA CARD" (registered trademark) virtual machine.